Learning Relative Order in Permutation EDAs: The ROSE Family for Single- and Multi-Objective Flow-Shop Scheduling

Warin Wattanapornprom

***Abstract*—Permutation estimation-of-distribution algorithms (EDAs) typically commit to one of two structural views of a candidate solution: an edge-histogram view, which learns immediate adjacency, or a node-histogram view, which learns absolute position. We introduce ROSE (Relative Order SEquencer), a family of six permutation EDA variants that instead learns the *signed displacement* between jobs, $\Delta_{ij} = pos(j) - pos(i)$: whether j tends to occur before or after i, and how far apart. The directed-edge relation is the special case $\Delta_{ij} = +1$; the classical order-schema relation ($i \prec j$) is the sign of Δ_{ij} alone. ROSE therefore adds a resolution — separation, not just order — that neither representation carries. We formalize three ways to turn this into a generative model (multi-reference averaging, single-reference parametric sampling, single-reference empirical histogram), each with and without punched-parent template inheritance (WT/WO), and evaluate all six on permutation flow-shop scheduling (Taillard instances) against makespan, total flow time, and machine idle time. In a matched 20-seed ablation (8400 runs, TA021–TA030), the same ranking — $ROSE$/WT $>$ SR-ROSE/WT $>$ SH-ROSE/WT $>$ ROSE/WO $>$ SR-ROSE/WO $>$ SH-ROSE/WO — holds in all three single objectives and all four multi-objective settings (Friedman $p < 10^{-170}$ throughout); template inheritance is simultaneously more accurate and roughly 40% faster than rebuilding every position from the model (rank-biserial effect $= 1.0$ in every WT-vs-WO comparison). Against a broader field of sixteen permutation EDAs (five matched seeds, TA001–TA030), ROSE/WT is mid-field on makespan and total flow time but the strongest or second-strongest algorithm on machine idle time in one third of the instance strata — a signal that is real but statistically fragile, and that depends on instance structure at least as much as on the algorithm. The interesting result is not that one algorithm wins every table — it does not — but that each objective, and each family of instances, exposes a different kind of learnable order. We report this as a confirmed family-internal result and a preliminary, objective-dependent field result, deliberately kept separate, and outline where an adaptive representation (our planned successor, CARE-EDA) would need to arbitrate between them.**

***Index Terms*—permutation estimation of distribution algorithm, relational order representation, flow-shop scheduling, template-guided sampling, generative optimization, signed pairwise displacement, multi-objective optimization**

I. Introduction

An estimation-of-distribution algorithm (EDA) is not a metaheuristic bolted onto a probability model; it *is* a probability model. Where a classifier predicts a label from a fixed, previously observed choice set, an EDA for combinatorial optimization must generate structured candidates — here, complete permutations of n jobs — that may never have appeared in the selected population it learned from, and it must do this every generation, under a shrinking evaluation budget. This is probabilistic, generative machine learning for optimization: an explicit distribution is fit to selected solutions, and new solutions are sampled from it. The question a permutation EDA answers first, before any question about search strategy, is representational: *what property of a permutation is the model allowed to see?*

Two answers dominate the literature. An edge-histogram model (EHBSA [1]) learns which jobs tend to sit immediately next to which other jobs — the directed adjacency $i \to j$. A node-histogram model (NHBSA [2]) learns which job tends to sit at which absolute position. Both are principled, both are cheap, and both discard information the other one keeps: an edge model cannot tell two solutions apart if neither places i and j adjacently, no matter how differently they place everything else about i and j's relative order; a position model must relearn the same relational pattern independently at every absolute coordinate it might occur at.

This paper introduces a third view. ROSE (Relative Order SEquencer) learns the *signed displacement* between two jobs,

$$\Delta_{ij} = pos(j) - pos(i), \tag{1}$$

directly: the sign of Δ_{ij} recovers the ordinal relation ($i \prec j$ iff $\Delta_{ij} > 0$), and its magnitude is separation, which neither the edge view nor the pure order-schema view represents. The directed edge $i \to j$ is recoverable as the single special case $\Delta_{ij} = +1$; it is not, and this paper does not claim it to be, a superset of everything an edge model can learn — only that the same (i, j) pair now carries a graded relation instead of a binary one.

Whether that extra resolution is worth its cost is an empirical question with, we will argue, an objective-dependent answer. Section II places ROSE in the lineage of representation-first thinking about permutation search, from Goldberg and Lingle's original diagnosis of why literal positional encoding fails on the travelling salesman problem [3] through order-based operators [4], [5] and forma analysis [6], [7] to the modern permutation-EDA literature [8]. Section IV formalizes the taxonomy sketched above. Section V specifies all six ROSE variants precisely against the actual implementation, not idealized pseudocode, flagging every place where documentation and code disagreed (§ V, and in full in `CLAIMS_AUDIT.md`).

The author is with the Applied Computer Science Program, Department of Mathematics, Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), Bangkok, Thailand. Affiliation confirmed by web verification against the author's published bioinformatics co-authorships during manuscript preparation; please re-check before submission.

Section VI works out why the histogram variant needs $O(n^3)$ storage while the other two remain $O(n^2)$. Sections VII–IX report three deliberately separate bodies of evidence at three deliberately different confidence levels, and Section X argues that the pattern across them — which objective rewards which representation, and on which instances — is the paper's actual contribution, more than any single win.

II. Related Work

A. From order schemata to relative-order learning

Holland's schema theorem describes building-block propagation for fixed-position, fixed-alphabet strings under low-disruption recombination. It does not transfer literally to permutations: a permutation is a bijection, so a schema like "$1{*}0{*}{*}1$" has no meaning once every symbol must appear exactly once. Goldberg and Lingle were the first to make this failure concrete for combinatorial optimization, diagnosing why direct positional encoding collapses on the travelling salesman problem and arguing that a permutation representation must be judged by what invariant it preserves under recombination, not by its superficial similarity to a binary string [3]. The order-based operators that followed — order crossover [4] and edge recombination [5] — are exactly two different answers to that question: OX preserves relative order (the sequence in which surviving jobs appear), edge recombination preserves adjacency. Neither is a special case of the other; they preserve genuinely different invariants. Radcliffe's forma analysis generalizes "schema" itself to an arbitrary equivalence class on any representation, giving a rigorous vocabulary for exactly this kind of comparison without forcing permutations into a binary-string frame [6], [7].

ROSE sits downstream of this lineage, not upstream of it: it is not a claim that schema theory extends to permutations, but a concrete answer, in that vocabulary, to what invariant a probabilistic model should learn. Position preserves $pos(i)$; the order-based lineage preserves $i \prec j$; edge recombination and its probabilistic descendant, EHBSA, preserve $i \to j$. ROSE preserves $\Delta_{ij} = pos(j) - pos(i)$, of which the directed edge is the $\Delta_{ij} = +1$ slice and the order relation is the sign. Table I makes this comparison concrete against every algorithm used later in this paper.

B. Estimation of distribution algorithms

EDAs replace the implicit, crossover-mediated preservation of building blocks with an explicit probabilistic model, fit to a selected subpopulation each generation and resampled from directly [9], [10], [11]. For permutation domains, the representational question above becomes the central design choice: EHBSA learns a directed adjacency histogram [1], later extended with local search and partial-solution (template) inheritance [12], [13]; NHBSA learns an independent position×node histogram [2]. Ceberio et al. [8] survey this space, including distance-based ranking models — Mallows models fit to a reference permutation and a dispersion parameter [14], [15] — random-key EDAs, which sample independent continuous keys and decode by sorting [16], and Plackett–Luce EDAs, which fit a single global worth parameter per item and sample sequentially without replacement [17], [18], [19]. All three are included as field baselines in Section VIII. The historical COIN lineage — positive/negative correlation learning applied first to a directed-edge representation [20] and then to positional and hybrid node/edge representations [21], [22], including the multi-objective extension in [23] — is this paper's own predecessor work and is treated in Section VIII as a representational competitor, not as ROSE's origin.

C. Multi-objective EDAs and reference-front evaluation

NSGA-II [24] and SPEA2 [25] established the elitist, nondominated-archive pattern this paper's MO variants inherit. Pelikan, Sastry, and Goldberg survey the specific risks a probabilistic model introduces in the multi-objective setting: diversity loss and premature convergence of the learned distribution onto one region of the front [26], a risk this paper's Discussion (§ X) returns to directly. Hypervolume and IGD$^+$ are reported with the modified, dominance-resistant distance calculation of Ishibuchi et al. [27]; both indicators are only ever compared within a fixed reference front and normalization, never across scopes with different pooling (§ VII-A).

D. Flow-shop scheduling beyond makespan

Taillard's benchmark instances [28] are the de facto standard for permutation flow-shop scheduling and are used here unmodified (TA001–TA030). Most EDA and metaheuristic evaluations on these instances report makespan alone; total flow time and machine idle time are comparatively neglected objectives that reward structurally different solutions [29], a distinction this paper's three objectives are chosen specifically to probe (§ III).

III. Problem Formulation and Objectives

An instance of the permutation flow-shop scheduling problem is a set of n jobs, each requiring processing on the same m machines in the same order; a candidate solution is a permutation π of the n jobs specifying the order in which they are released onto the first machine (and, because machine order is shared, onto every subsequent machine as well). Let $C_{i,k}(\pi)$ denote the completion time of the job in position i of π on machine k. We report three objectives, computed by direct simulation of the schedule implied by π:

- **Makespan**, $C_{\max}(\pi) = C_{n,m}(\pi)$: the completion time of the last job on the last machine. Makespan is dominated by the *critical path* through the schedule — the single longest chain of dependent operations — so it rewards getting a small number of bottleneck placements right and is comparatively insensitive to the rest of the permutation.
- **Total flow time**, $\sum_{i=1}^{n} C_{i,m}(\pi)$: the sum of every job's completion time on the last machine. Unlike makespan, this objective is sensitive to *every* job's position, and specifically rewards placing many jobs early, not just avoiding one late bottleneck.
- **Machine idle time**: the total time machines spend idle while waiting for a job to become available, summed across all m machines. This depends on placement *and*

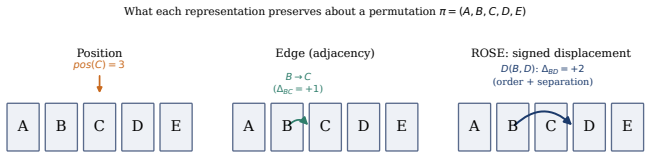

Fig. 1. What each representation preserves about a permutation.

on relational compatibility between consecutive jobs' processing-time profiles across machines, not on either job's absolute position or the makespan-defining critical path alone.

These three objectives are chosen because they plausibly expose different kinds of learnable structure: a model that only sees adjacency or only sees position has no obvious advantage for total flow time (which is a sum over all positions, not one path); a model that sees signed relative order has, in principle, more to say about idle time, which depends on how the processing-time profile of one job's neighbor compares to its own. Section VIII tests whether that intuition survives contact with data. Objectives are always reported and optimized independently (single-objective runs) or jointly via Pareto dominance (multi-objective runs, Section IX); we never combine them into a single weighted scalar.

IV. Representation Taxonomy

Figure 1 lays out the three representations formally: given $\pi = (A, B, C, D, E)$, a position model asks "at what index does C sit" ($pos(C) = 3$); an edge model asks "does B immediately precede C" ($\Delta_{BC} = +1$); ROSE asks the graded version of the same question, "how far, and in which direction, is C from B" ($\Delta_{BC} = +2$ here). Table I extends this to every algorithm compared in this paper, including the storage each representation needs and information each necessarily discards.

Two representations that look similar can behave very differently. Consider $\pi_1 = (A, B, C, D, E, F)$ and $\pi_2 = (A, D, E, F, B, C)$ (Figure 2). Both satisfy the order relation $A \prec C$; neither places A and C adjacently, so an edge model sees no edge $A \to C$ in either permutation and cannot distinguish them; a pure order-schema view sees $A \prec C$ in both and, again, cannot distinguish them. Yet $\Delta_{AC} = +2$ in π_1 and $\Delta_{AC} = +5$ in π_2 — two structurally very different placements that only the signed-displacement view separates.

This resolution is not free. A model over Δ_{ij} has, in the worst case, $n - 1$ distinguishable values per ordered pair instead of a binary (edge model) or single-scalar (position model) quantity, which is exactly why Section VI spends real attention on what each ROSE variant actually stores.

V. The ROSE Family

All six variants share one estimator and one construction loop; they differ in how many reference jobs a target position is predicted from, and whether that prediction is a single scalar draw, an average over several predictions, or a full empirical lookup. Every claim

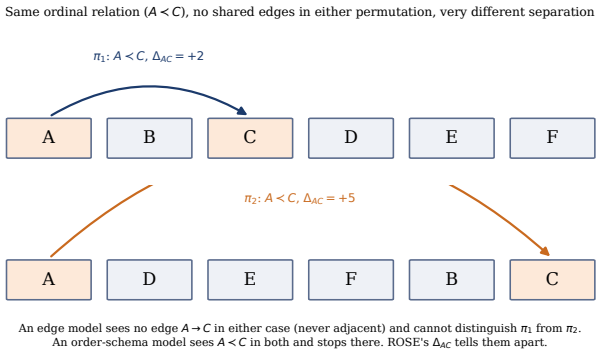

Fig. 2. Two permutations that agree on order and share no edges, but differ in separation — exactly what Δ_{ij} adds over both weaker representations.

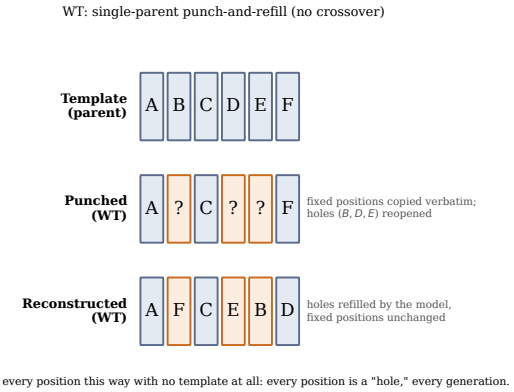

Fig. 3. WT reconstructs only the punched holes of one template parent; WO treats every position as a hole, every generation.

in this section was checked against the implementation (`src/coin/models/rose.py`, `rose_histogram.py`, `hbsa.py`) directly, not against its docstrings; every disagreement found between code and documentation is recorded verbatim in `CLAIMS_AUDIT.md`.

A. Naming: WO and WT

WO ("without template") variants build every position of every offspring from the model, every generation. **WT** ("with template") variants instead select one elite individual as a *template*, reopen a random subset of its positions as "holes" (a Fisher–Yates partial shuffle, shared verbatim with EHBSA's and NHBSA's own WT mode [12]), copy every other position unchanged, and reconstruct only the holes from the model (Figure 3). This is single-parent punch-and-refill: there is no second parent and no recombination operator anywhere in this code path, so we do not call it crossover.

B. Model estimation

From the selected (truncation, top-k) subpopulation, every variant fits, for every ordered pair (i, j) of jobs, the node-position marginal

$$N[j, p] = \frac{1}{|S|} \sum_{\pi \in S} \mathbb{1}[\pi(p) = j], \tag{2}$$

TABLE I
REPRESENTATION COMPARISON. "DEPENDENCE CAPTURED" IS THE STRUCTURAL RELATIONSHIP EACH MODEL'S SAMPLING STEP CAN CONDITION ON; "INFORMATION DISCARDED" IS WHAT A MODEL OF THAT FAMILY CANNOT REPRESENT EVEN IN PRINCIPLE, INDEPENDENT OF SAMPLE SIZE. STORAGE IS PER-MODEL, NOT PER-SAMPLE.

Algorithm	Representation	Sampling mechanism	Dependence captured	Information discarded	Storage	Objective affinity (this study)
EHBSA/WO, WT	Directed adjacency	Edge-histogram walk	Immediate successor $i \to j$	Separation beyond 1 step	$O(n^2)$	Makespan (Fig. 6)
NHBSA/WO, WT	Absolute position	Position × node histogram	$pos(i)=k$	All pairwise relations	$O(n^2)$	Makespan, flow time (Fig. 6)
COIN, COIN/WT	Signed directed edge	Reward/punishment edge learning	$i \to j$, sign-weighted	Separation beyond 1 step	$O(n^2)$	Weak across the board (Fig. 6)
NB-COIN, NB-COIN/WT	Signed position	Reward/punishment position learning	$pos(i)=k$, sign-weighted	All pairwise relations	$O(n^2)$	Flow time (Fig. 6)
SNE-COIN, HNE-COIN, CNE-COIN, CNB-COIN	Hybrid node/edge blends	Blended edge+position update	Both of the above, blended	Relative separation not on the blend's own axis	$O(n^2)$	Mid-field, instance-dependent (Fig. 6)
GM-EDA	Consensus permutation	Kendall-distance dispersion around a mode	Global similarity to one consensus ranking	Local, pairwise relational detail	$O(n)$	Mid-to-low field (Fig. 6)
RK-EDA	Independent per-item keys	Gaussian random keys, sort to decode	Marginal rank tendency per item	Any inter-item coupling	$O(n)$	Makespan (Fig. 6), fragile on idle time (Table VII)
PL-EDA	Global worth parameters	Plackett–Luce sequential sampling	Global preference order	Local/relative-position detail	$O(n)$	Weakest field-wide (Fig. 6)
ROSE/WO, WT (this paper)	Signed relative displacement Δ_{ij}	Multi-reference averaged target-position prediction	Signed order *and* separation, per ordered pair	Absolute position identity	$O(n^2)$	Idle time (Table VII), family-internal winner (Table III)
SR-ROSE/WO, WT (this paper)	Signed relative displacement Δ_{ij}	Single-reference truncated-normal draw	Same as ROSE, one reference only	Multimodal Δ distributions (unimodal assumption)	$O(n^2)$	Second within family (Table III)
SH-ROSE/WO, WT (this paper)	Signed relative displacement Δ_{ij}	Single-reference empirical histogram lookup	Same as ROSE, full empirical Δ distribution	Nothing structurally, but data-hungry at this budget	$O(n^3)$	Third within family, expressiveness unrewarded at this sample size (Table III, §Discussion)

together with four signed-displacement statistics, $\bar{\Delta}_{ij}$, σ_{ij}, $\Delta_{ij}^{\min}$, and $\Delta_{ij}^{\max}$ — the sample mean, standard deviation, minimum, and maximum of $pos_\pi(j) - pos_\pi(i)$ over $\pi \in S$. A fifth stored array, `distance_count`, is a constant ($|S|$ off-diagonal, 0 on the diagonal) rather than a genuine per-pair sample count, because every complete selected permutation supplies a valid displacement observation for every $i \neq j$ pair; there is no sparsity to count, and the paper treats this array as a fitted/unfitted flag, not an informative statistic (a discrepancy between the module's own docstring, which describes "four" statistics, and the actual five stored arrays plus N, recorded in `CLAIMS_AUDIT.md`). Every complete model is therefore six $(n \times n)$ arrays, regardless of variant — the basis for the $O(n^2)$ storage claim in Section VI. SH-ROSE additionally builds a full empirical tensor over Δ (§ VI).

C. Reference selection and target-position prediction

Construction proceeds one job at a time, in a randomized order, over a shrinking pool of free positions. For each job to be placed, one or more already-placed jobs are drawn as *references* from a randomized-length trailing window of the construction order so far.

ROSE (multi-reference). Every informative reference in the window predicts its own target position, $pos(\text{ref}) + \bar{\Delta}_{\text{ref},j}$; the predictions are averaged, and every free position is scored by an inverse-distance-to-target term blended with the node marginal $N[j, \cdot]$, then sampled by softmax. This is the only variant that combines more than one reference's opinion.

SR-ROSE (single-reference, truncated normal). Exactly one reference is drawn; a displacement is sampled from a truncated normal fit to $(\bar{\Delta}_{\text{ref},j}, \sigma_{\text{ref},j})$ via Box–Muller rejection over $[\Delta^{\min}, \Delta^{\max}]$, explicitly excluding $\Delta = 0$ (structurally impossible in a permutation, since no two jobs share a position, but defensively guarded against anyway at three separate points in the code, not merely once). The target position is $pos(\text{ref}) + \Delta$, scored exactly as in ROSE.

SH-ROSE (single-reference, empirical histogram). Exactly one reference is drawn; instead of a parametric draw, every free position's score comes directly from the empirical probability $\hat{P}(\Delta_{\text{ref},j} = \delta)$ read out of a full histogram over δ (§ VI), with the δ=0 bin forced to zero regardless of any (structurally impossible) count landing there.

D. Feasibility

Feasibility is structural, not repaired. Each job is drawn from, and removes itself from, a shrinking pool of free positions; because a position can only be claimed once and every job claims exactly one position, no duplicate placement or unfilled position can occur by construction. The only feasibility-related code is a defensive assertion after construction completes, never a corrective step.

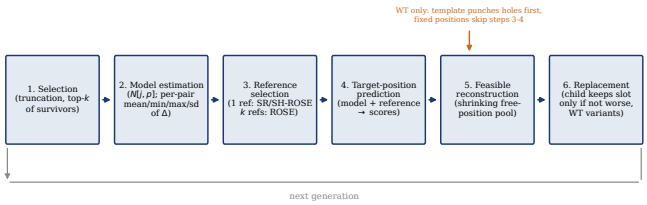

Fig. 4. One ROSE generation cycle. WT variants punch a template's holes before step 3 and skip steps 3–4 for every fixed position.

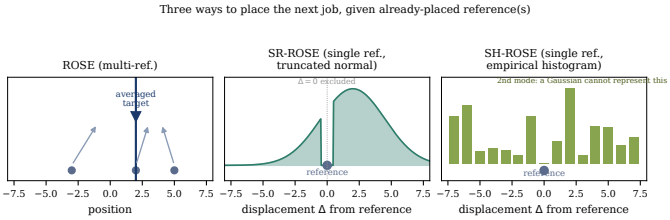

Fig. 5. Three ways to predict a target position from already-placed references. Only SH-ROSE's empirical histogram can represent a genuinely multimodal Δ distribution (right).

E. Selection and replacement

Truncation selection re-fits the model from scratch every generation (no incremental or exponentially-decayed update). For WT variants specifically, an additional elitist rule applies at replacement: an offspring replaces its own template only if it is not worse, a (1+1)-style ratchet per template slot that WO variants do not have. Section VIII finds this ratchet is, empirically, most of the reason WT keeps improving after generation 200 while WO plateaus or regresses (Figure 7).

VI. Complexity Analysis

Compact ROSE (ROSE, SR-ROSE, and both of their WT variants) stores exactly six $(n \times n)$ arrays — N, $\bar{\Delta}$, $\Delta^{\min}$, $\Delta^{\max}$, σ_Δ, and the fitted-flag array — giving $O(n^2)$ memory, confirmed directly from the dataclass field shapes in `rose.py` and from `diagnostics()`'s own byte-count. SH-ROSE additionally allocates a full tensor

$$H \in \mathbb{Z}^{n \times n \times (2n-1)}, \tag{3}$$

indexed by (i, j, Δ) with $\Delta \in \{-(n-1), \ldots, -1, 1, \ldots, n-1\}$ (no zero bin), which is $O(n^3)$ because the third axis is itself $O(n)$. This is confirmed both from the allocation itself (`counts = np.zeros((n, n, 2n-1))`) and from a unit test that asserts the exact shape for a known n. The practical consequence, tested directly in Section VIII, is that SH-ROSE's extra expressiveness — representing an arbitrarily multimodal Δ distribution instead of a single Gaussian mode (Figure 5, right panel) — comes at roughly n times the memory of SR-ROSE for the same information, and at this paper's sample size and generation budget, that expressiveness is not rewarded: SH-ROSE ranks behind SR-ROSE in every single objective and every multi-objective setting tested (Table III).

TABLE II
THE THREE EVIDENCE SCOPES. EACH ANSWERS A DIFFERENT QUESTION AND CARRIES A DIFFERENT NUMBER OF SEEDS.

Scope	Instances	Seeds	Algorithms
A: full-field	TA001–TA030	5 (42, 2026–2029)	16 (field)
B: convergence	TA001–TA020	10 (42, 2026–2034)	16 (field)
C: family ablation	TA021–TA030	20 (42, 2026–2044)	6 (ROSE only)

VII. Experimental Design

All experiments use Taillard's permutation flow-shop instances [28], TA001–TA030 (20 jobs, 5 machines, per the standard Taillard naming for this instance group). Three deliberately separate evidence scopes are used throughout this paper and are never merged (Table II):

Scope A (full-field comparison) answers whether ROSE competes against other representations at all: five matched seeds, sixteen algorithms (Table I), reported separately for TA001–010, TA011–020, TA021–030, and the TA001–030 pool (Figure 9). It has the fewest seeds of the three scopes and is never presented as more than preliminary.

Scope B (convergence checkpoint) diagnoses learning speed and plateaus: ten seeds, TA001–TA020 only (*not* TA021–030, because this run family does not cover it), all sixteen trajectories, generations 1–400.

Scope C (confirmed ROSE-family ablation) establishes ordering within the ROSE family only: TA021–TA030, seeds 42 and 2026–2044 (twenty seeds), population 100, 400 generations, six ROSE variants, evaluation budget 40,000 candidates per run. This scope produced 8400 completed runs and zero recorded errors (10 instances × 20 seeds × 6 variants × 7 objective settings = 8400; three single objectives plus four multi-objective settings). It does *not* establish superiority over the field of sixteen, over RK-EDA specifically, over COIN variants, or over GA/NSGA-II/SPEA2-style search — only ordering inside the ROSE family.

Every configuration uses population 100, truncation selection ratio 10% (ROSE variants) with one untimed warm-up generation excluded from timing, and a fixed deterministic-parity check before every timed run. Table I of the reproducibility checklist (`README.md`) lists exact software versions and reproduction commands.

A. Metric definitions and safeguards

Relative percentage deviation is defined per block (one instance × one seed, within one scope) against the pooled best observed inside that scope:

$$RPD(a, i, s) = 100 \cdot \frac{f(a, i, s) - f_{\text{best}}(i, s)}{\max(f_{\text{best}}(i, s),\ 1)}. \tag{4}$$

The $\max(\cdot, 1)$ floor matters only for machine idle time, where 10 of 150 Scope A blocks (both instances TA002 and TA008, every one of their five matched seeds) have a block-best of exactly zero idle time; dividing by zero there would make RPD undefined for every algorithm in that block. We reverse-engineered this exact floor policy from the source pipeline's published numbers (it reproduces `stratified-rpd.json` to four decimal places once applied) rather than inventing our

own; we report raw regret and rank alongside RPD everywhere this floor is used (Table VII), and never trust RPD alone when the reference is unstable, per the master specification for this study. Wins are ties broken by exact equality with the block minimum; ranks are average ranks under ties.

For multi-objective results, hypervolume and IGD$^+$ are taken from the Scope C pipeline's own aggregate file, computed against a pooled observed nondominated front per instance×seed block — normalized and comparable within Scope C, and never compared against a differently-pooled front from another scope. We did not re-derive block-level HV/IGD$^+$ ourselves for a paired significance test, because doing so would require reconstructing that exact pooling and normalization procedure and risks introducing a silently different reference front (§ XI, `EXPERIMENT_GAPS.md`); where the master specification allows marking a test as pending rather than reconstructing it from aggregates, we do so. Archive size and frontier contribution (the fraction of a pooled per-block nondominated front contributed by one variant) were instead recomputed directly from the raw per-run archives for Figure 11, since both are well-defined without any reference-front choice.

B. Statistical analysis

Instance×seed is the paired block throughout. For Scope C, a Friedman omnibus test ($N = 200$ blocks, $k = 6$ variants) is run per objective, followed by two-sided Wilcoxon signed-rank tests of every variant against ROSE/WT, Holm-corrected within each objective, with matched-pairs rank-biserial effect sizes (Table IV, Table V). The TA strata are never pooled across Scope A and Scope C or treated as one experiment. Generations within one run are not treated as independent observations anywhere in this paper.

VIII. Single-Objective Results

A. Confirmed: the ROSE-family ablation (Scope C)

The same ranking, $ROSE/WT > SR-ROSE/WT > SH-ROSE/WT > ROSE/WO > SR-ROSE/WO > SH-ROSE/WO$, holds in every one of the seven objective settings tested (Table III, and the four multi-objective settings in Table VIII), confirmed by a Friedman omnibus test with $p < 10^{-170}$ in every single objective (Table IV) and by pairwise Wilcoxon tests against ROSE/WT that are all significant after Holm correction at $p_{Holm} < 10^{-12}$ (Table V). Every WT-vs-WO comparison has rank-biserial effect $r_{rb} = 1.00$: ROSE/WT beats every WO variant in literally every non-tied block, a complete separation, not a marginal statistical edge.

Figure 6 makes the mechanism visible: WO variants are not merely slower, their mean best-of-generation value is essentially flat and noisy from early in the run. ROSE/WT's mean gain from generation 200 to 400 is $+0.84\%$ (makespan), $+0.70\%$ (total flow time), and $+3.60\%$ (idle time) — run to completion but not converged, especially for idle time, where the second half of the run still closes more than a twentieth of the remaining gap. WO variants, lacking WT's per-template elitist ratchet (§ V), show a materially different pattern in the same window: makespan gain is slightly *negative* for all

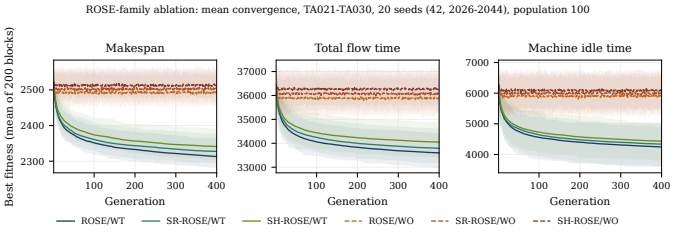

Fig. 6. Mean convergence, Scope C (20 seeds, TA021–030, population 100). WT variants (solid) keep descending; WO variants (dashed) plateau at a visibly noisy level from early generations onward.

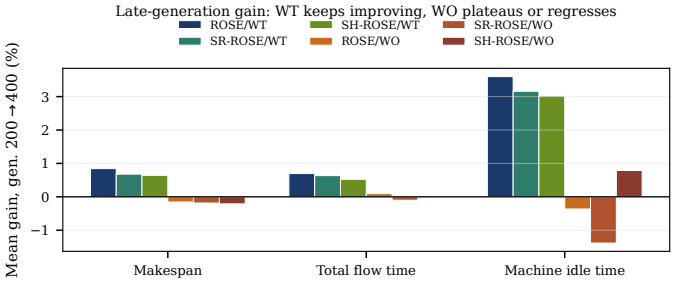

Fig. 7. Mean relative gain from generation 200 to 400. WT variants continue to improve on all three objectives; WO variants plateau or measurably regress on makespan and (for two of three variants) on idle time.

three WO variants (-0.15% to -0.21%); flow time is roughly flat (ROSE/WO $+0.10\%$, the other two slightly negative); idle time is mixed (ROSE/WO -0.36%, SR-ROSE/WO -1.38%, SH-ROSE/WO $+0.79\%$) (Figure 7, full values in Table III). We read this as direct evidence that WT's advantage is not merely computational — reconstructing fewer positions per generation — but structural: the per-template replacement rule gives WT a ratchet against regression that WO does not have.

Mean runtime per run is 29.3–31.4 s across the three WT variants and 49–60 s across the three WO variants — but that 54–60 second figure is a *cross-variant* span (ROSE/WO ≈49–52 s, up to SH-ROSE/WO ≈57–60 s), not the spread of any one WO variant across objectives (Table III, Figure 8). WT reconstructs only the punched holes of a template each generation, which is both why it is faster and, per the replacement-ratchet argument above, part of why it is also more accurate; we report this as an empirical runtime result specific to this implementation and these settings, not a universal complexity proof.

B. Preliminary: the full field (Scopes A and B)

Against the field of sixteen, ROSE/WT is not a leaderboard winner on makespan or total flow time: it ranks 5th of 16 on makespan (mean RPD 1.35%, behind NHBSA/WT, RK-EDA, NB-COIN/WT, and NB-COIN) and 7th of 16 on total flow time (mean RPD 1.41%, essentially tied with COIN/WT's 1.41%, but with only 1 outright win in 150 blocks). On machine idle time, ROSE/WT ranks 2nd of 16 by mean RPD (19.7%, against NB-COIN/WT's 19.0% and well ahead of NHBSA/WT's 26.2%) at the pooled TA001–030 level — but this pooled number hides a sharp stratum effect (Figure 9): ROSE/WT's idle-time strength is concentrated almost entirely

TABLE III
SCOPE C (TA021–TA030, 20 SEEDS, POPULATION 100, 400 GENERATIONS): SINGLE-OBJECTIVE RESULTS, ROSE-FAMILY ABLATION. RPD IS AGAINST THE POOLED FAMILY BEST FOR THAT INSTANCE×SEED BLOCK. FRIEDMAN OMNIBUS TEST PER OBJECTIVE, $N = 200$ BLOCKS, $k = 6$ VARIANTS (TABLE IV).

Objective	Variant	Mean RPD (%)	Wins/200	Mean rank	Gain gen. 200→400 (%)	Mean runtime (s)
Makespan	ROSE/WT	0.108	151	1.29	0.843	29.85
	SR-ROSE/WT	0.726	41	2.00	0.679	30.91
	SH-ROSE/WT	1.329	10	2.72	0.643	31.21
	ROSE/WO	3.469	0	4.57	-0.155	51.60
	SR-ROSE/WO	3.816	0	4.97	-0.185	53.51
	SH-ROSE/WO	4.210	0	5.45	-0.211	59.61
Total flow time	ROSE/WT	0.031	174	1.13	0.697	30.02
	SR-ROSE/WT	0.600	25	1.93	0.634	30.79
	SH-ROSE/WT	1.386	1	2.96	0.524	31.39
	ROSE/WO	3.219	0	4.49	0.095	52.09
	SR-ROSE/WO	3.541	0	4.92	-0.101	54.07
	SH-ROSE/WO	3.965	0	5.58	-0.005	59.81
Machine idle time	ROSE/WT	0.763	139	1.37	3.598	29.29
	SR-ROSE/WT	3.004	48	2.01	3.160	30.32
	SH-ROSE/WT	5.543	13	2.63	3.020	30.70
	ROSE/WO	16.933	0	4.67	-0.363	51.19
	SR-ROSE/WO	18.504	0	5.05	-1.384	52.93
	SH-ROSE/WO	19.654	0	5.25	0.788	58.69

TABLE IV
SCOPE C STATISTICAL TESTS. FRIEDMAN OMNIBUS ACROSS THE 6 VARIANTS (N=200 BLOCKS); PAIRWISE TWO-SIDED WILCOXON SIGNED-RANK VS. ROSE/WT, HOLM-CORRECTED ($\alpha = 0.05$), WITH MATCHED-PAIRS RANK-BISERIAL EFFECT SIZE r_{rb}.

Objective	$\chi^2(5)$	p	Kendall's W
Makespan	849.3	2.50×10^{-181}	0.849
Total flow time	895.8	2.15×10^{-191}	0.896
Machine idle time	821.5	2.56×10^{-175}	0.822

TABLE V
PAIRWISE TWO-SIDED WILCOXON SIGNED-RANK TESTS, ROSE/WT VS. EACH OTHER VARIANT, HOLM-CORRECTED WITHIN EACH OBJECTIVE (N=200 PAIRED BLOCKS). CELL FORMAT: p_{Holm} (RANK-BISERIAL EFFECT r_{rb}); ALL 15 COMPARISONS ARE SIGNIFICANT AT $p_{Holm} < 10^{-12}$.

vs. ROSE/WT	Makespan	Flow time	Idle time
SR-ROSE/WT	5.4×10^{-20} (0.55)	8.4×10^{-28} (0.75)	1.3×10^{-13} (0.48)
SH-ROSE/WT	2.7×10^{-31} (0.88)	7.2×10^{-34} (0.99)	7.4×10^{-28} (0.78)
ROSE/WO	7.1×10^{-34} (1.00)	7.2×10^{-34} (1.00)	7.2×10^{-34} (1.00)
SR-ROSE/WO	7.1×10^{-34} (1.00)	7.2×10^{-34} (1.00)	7.2×10^{-34} (1.00)
SH-ROSE/WO	7.1×10^{-34} (1.00)	7.2×10^{-34} (1.00)	7.2×10^{-34} (1.00)

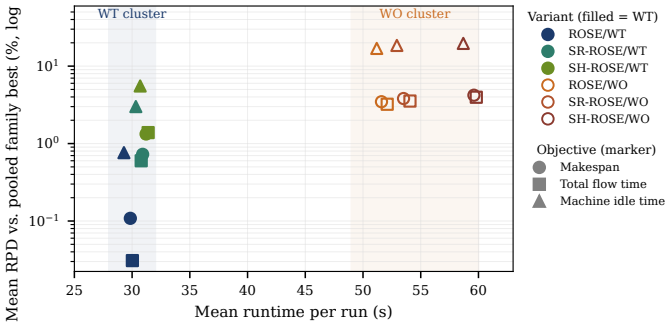

Fig. 8. Runtime/quality trade-off, Scope C. WT and WO form two clearly separated clusters; WT is both faster and more accurate.

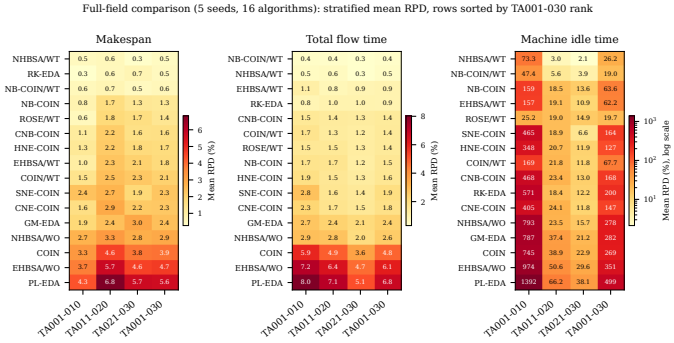

Fig. 9. Stratified mean RPD, Scope A (5 seeds, 16 algorithms). Rows sorted by TA001–030 mean rank. Machine idle time uses a log color scale because of the zero-reference-block instability discussed in §VII-A.

in TA001–010, where it wins the RPD comparison outright (25.2% vs. NB-COIN/WT's 47.4% and NHBSA/WT's 73.3%, and takes 34 of 50 blocks outright) but does not appear in the top five of either TA011–020 or TA021–030, where NHBSA/WT and NB-COIN/WT take over. Whatever ROSE is picking up on idle time is at least as much a property of *which instances* as of the objective in the abstract.

Scope B (ten seeds, TA001–020) sharpens this further, and is the clearest illustration in this paper of why RPD, wins, and raw regret can disagree. By mean RPD, ROSE/WT ranks first of sixteen on idle time (29.2%); by win count, it ranks third (62 of 200 blocks, behind NHBSA/WT's 121 and NB-COIN/WT's 96); by mean raw regret, it also ranks third (63.8, behind NHBSA/WT's 9.8 and NB-COIN/WT's 21.6) (Table VII). Nineteen of the 200 blocks in this scope have a zero reference, and RPD's floor policy (§VII-A) rewards an algorithm disproportionately for doing well specifically on those extreme blocks. ROSE/WT's RPD-based "first place" on idle time is real but is the least robust of the three lenses;

TABLE VI
SCOPE A FULL-FIELD COMPARISON, TA001–030 POOLED (5 SEEDS, 16 ALGORITHMS, N=150 BLOCKS). ALL COLUMNS ARE MEAN RPD (%); IDLE TIME USES A FLOORED DENOMINATOR (§METRIC DEFINITIONS). ROSE/WT IS HIGHLIGHTED.

Algorithm	Makespan (%)	Flow time (%)	Idle time (%)
NHBSA/WT	0.46	0.47	26.2
RK-EDA	0.51	0.93	200.4
NB-COIN/WT	0.60	0.38	19.0
NB-COIN	1.27	1.54	63.6
ROSE/WT	**1.35**	**1.41**	**19.7**
CNB-COIN	1.62	1.38	168.0
HNE-COIN	1.75	1.57	126.8
EHBSA/WT	1.80	0.92	62.2
COIN/WT	2.05	1.41	67.7
SNE-COIN	2.31	1.95	163.6
CNE-COIN	2.25	1.83	147.1
GM-EDA	2.42	2.41	281.7
NHBSA/WO	2.91	2.56	277.5
COIN	3.90	4.80	269.0
EHBSA/WO	4.68	6.07	351.4
PL-EDA	5.61	6.75	498.6

TABLE VII
SCOPE B (TA001–020, 10 SEEDS, N=200 BLOCKS), MACHINE IDLE TIME: THREE VIEWS OF THE SAME 16-ALGORITHM FIELD. RANK 1 = BEST. 19/200 BLOCKS HAVE A ZERO REFERENCE (RPD FLOORED, §METRIC DEFINITIONS), WHICH IS WHY THE RPD-BASED AND WIN/REGRET-BASED LEADERS DISAGREE.

Algorithm	RPD rank	Win rank	Regret rank
ROSE/WT	1	3	3
NHBSA/WT	2	1	1
NB-COIN/WT	3	2	2
EHBSA/WT	4	7	5
NB-COIN	5	4	4
COIN/WT	6	9	8

NHBSA/WT is the more typical, lower-variance performer by the other two. We report all three, deliberately, rather than the one that reads best.

IX. MULTI-OBJECTIVE RESULTS

Within the family ablation, ROSE/WT leads mean hypervolume and mean IGD$^+$ in all four multi-objective settings (Table VIII), and wins the per-block hypervolume comparison in a clear majority of blocks in every setting (126–129 of 200 for the two-objective settings, 111 of 200 for three objectives). Frontier contribution — the share of each block's pooled nondominated front contributed by one variant — ranges from 36.8% (three objectives, where six variants must share the front) to 50.0% (makespan×flow time, where ROSE/WT alone supplies half of every pooled front on average). Mean archive size grows with the number of objectives, as expected (from roughly 7.5 for two-objective settings to 42.5 for three), without changing the family ordering. None of this establishes superiority over NSGA-II, SPEA2, or the sixteen-algorithm field in the multi-objective setting; Scope C's MO runs never included a non-ROSE competitor.

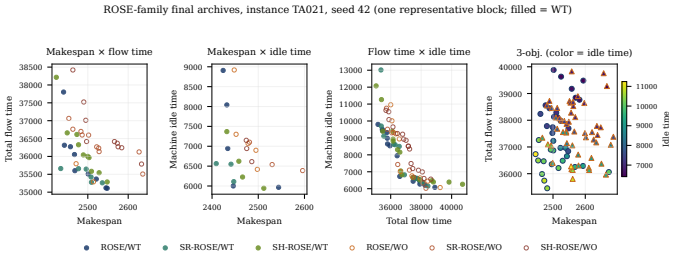

Fig. 10. Final ROSE-family archives, one representative block (TA021, seed 42). WT variants (filled) form a visibly tighter, better-positioned front than WO (open) in every objective pair.

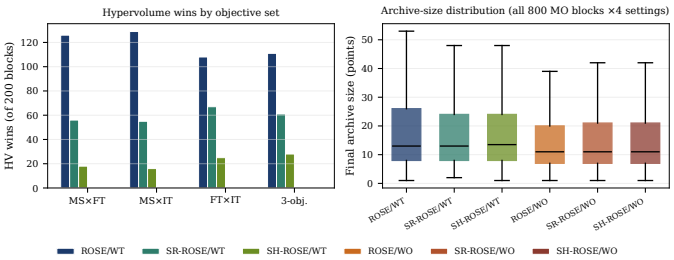

Fig. 11. Hypervolume wins (left) and per-block archive-size distribution (right), Scope C. Archive size is a directly-measured quantity requiring no reference-front choice.

X. DISCUSSION

The three objectives in this study reward structurally different things, and the results above are consistent with that, not merely decorated by it. Makespan is set by a single critical path; a model that gets the few bottleneck placements right does most of the work, which is plausibly why position-sensitive and edge-sensitive models (NHBSA/WT, RK-EDA, NB-COIN/WT) lead the field there and ROSE/WT is mid-pack. Total flow time sums over every job's completion time, rewarding many small correct decisions rather than a few large ones; ROSE/WT is competitive but not distinguished here either. Machine idle time depends on the relational compatibility between *consecutive* jobs' processing-time profiles across machines, which is closer to what a relative-displacement model is built to see — and this is exactly the one objective where ROSE/WT's signal shows up, concentrated specifically in the TA001–010 instance family. The honest reading is not "ROSE is good at idle time" but "some instances' idle-time structure is learnable through relative order, and TA002/TA008-like instances in particular reward it strongly enough to show up even through an unstable metric."

Inside the family, the WT/WO contrast is the paper's cleanest finding: WT is not a free lunch that happens to also be faster, it is a different learning dynamic. The per-template elitist replacement rule (§ V) gives every offspring a floor — it cannot be worse than the template it came from — which WO's from-scratch reconstruction never provides. That this shows up as a rank-biserial effect of exactly 1.00 in every WT-vs-WO comparison, and as visibly negative mean late-generation gain for WO on makespan specifically, argues that template inheritance is doing real algorithmic work, not just saving compute.

TABLE VIII
SCOPE C MULTI-OBJECTIVE RESULTS, ROSE-FAMILY ABLATION. HV/IGD$^+$ COMPUTED BY THE SOURCE PIPELINE AGAINST A POOLED OBSERVED NONDOMINATED FRONT PER INSTANCE×SEED BLOCK (NOT RE-DERIVED IN THIS AUDIT – SEE §STATISTICAL ANALYSIS).

Objective set	Variant	Mean HV	HV wins/200	Mean IGD$^+$	IGD wins/200	Frontier contrib. (%)	Archive size
Makespan × flow time	ROSE/WT	1.0226	126	0.0308	140	50.0	7.53
	SR-ROSE/WT	0.9784	56	0.0569	50	31.5	7.60
	SH-ROSE/WT	0.9226	18	0.0897	10	16.4	7.49
	ROSE/WO	0.7216	0	0.2144	0	1.1	6.40
	SR-ROSE/WO	0.6694	0	0.2512	0	0.7	6.54
	SH-ROSE/WO	0.6219	0	0.2864	0	0.4	6.55
Makespan × idle time	ROSE/WT	1.0243	129	0.0286	134	49.2	9.34
	SR-ROSE/WT	0.9850	55	0.0489	48	28.6	9.36
	SH-ROSE/WT	0.9591	16	0.0642	17	18.9	9.04
	ROSE/WO	0.8014	0	0.1579	1	1.6	8.09
	SR-ROSE/WO	0.7668	0	0.1795	0	1.1	7.90
	SH-ROSE/WO	0.7253	0	0.2076	0	0.6	8.12
Flow time × idle time	ROSE/WT	0.9634	108	0.0203	126	40.4	17.53
	SR-ROSE/WT	0.9536	67	0.0268	55	29.6	17.09
	SH-ROSE/WT	0.9365	25	0.0342	19	22.6	17.01
	ROSE/WO	0.8435	0	0.0836	0	3.0	14.42
	SR-ROSE/WO	0.8287	0	0.0925	0	2.5	14.45
	SH-ROSE/WO	0.8162	0	0.0985	0	2.0	14.26
Three objectives	ROSE/WT	0.9405	111	0.0273	120	36.8	42.53
	SR-ROSE/WT	0.9238	61	0.0334	65	29.3	42.35
	SH-ROSE/WT	0.9020	28	0.0416	15	23.3	42.13
	ROSE/WO	0.7696	0	0.0921	0	4.7	34.91
	SR-ROSE/WO	0.7425	0	0.1048	0	3.2	35.52
	SH-ROSE/WO	0.7304	0	0.1119	0	2.7	33.60

The ordering SR-ROSE > SH-ROSE within each WT/WO pair is the paper's most conceptually interesting negative result. SH-ROSE's empirical histogram can, in principle, represent a genuinely multimodal displacement distribution that a single Gaussian mode (SR-ROSE) cannot (Figure 5); at this study's sample size and generation budget, that extra expressiveness is not rewarded, and SH-ROSE pays roughly $n\times$ the memory of SR-ROSE (§ VI) for a representation that a truncation-selected population of 10–20 individuals per generation may simply not populate densely enough to estimate reliably. ROSE's relative-order information is promising but data-hungry; the histogram variant is the clearest illustration of that cost not yet being paid off.

A model that learns solutions as too similar risks collapsing diversity specifically in multi-objective search, where the value of an archive is in its spread, not its average quality [26]. This study's MO results do not directly test for collapse — archive size (Figure 11, right) is a coarse proxy at best — but the fact that ROSE/WT's frontier contribution is highest precisely in the setting with the smallest archives (two-objective, ∼50%) and lowest in the setting with the largest (three-objective, ∼37%, shared among six variants) is at least consistent with a family that concentrates rather than spreads, worth checking directly in future work.

Taken together, these results argue against building one universal permutation representation and for a system that can tell which representation an objective (and an instance family) rewards. That is precisely the premise of this paper's planned successor, CARE-EDA, which we intend to adapt among edge, position, and relational representations rather than committing to one in advance.

XI. THREATS TO VALIDITY

Statistical archiving. The Scope C source files retain median with quartiles or median with min/max per cell, not all 20 raw repetitions; Section VII's Friedman/Wilcoxon tests operate on the raw per-block *final* values (which are archived per run), not on a reconstructed within-cell distribution, so they are valid paired tests across blocks but say nothing about within-block seed-to-seed noise beyond what 20 independent seeds already provide.

Zero-reference RPD instability. Every RPD number reported for machine idle time inherits the floor policy of §VII-A; we mitigate this by always reporting wins and raw regret alongside RPD for that objective, but any future work that recomputes RPD without the same floor will not reproduce our idle-time numbers exactly, by design (§VII-A documents the floor precisely enough to reproduce it).

MO reference-front risk. We deliberately did not attempt to reconstruct block-level HV/IGD$^+$ ourselves (§VII-A); the paired significance test for the multi-objective results is therefore pending, not negative, and is listed as such in `EXPERIMENT_GAPS.md` rather than silently omitted.

Scope C tests ordering, not superiority. Every statistical test in Section VIII compares ROSE variants against each other. None of them compares ROSE against RK-EDA, COIN, NHBSA, GA-OX, NSGA-II, or SPEA2; that comparison exists only in Scope A/B, at five and ten seeds respectively, and is reported as preliminary throughout.

Instance and generalization scope. All results are on Taillard's 20-job, 5-machine instance family; nothing here has been tested on a second permutation problem, a different instance size, or a different generation budget. The idle-

time signal's concentration in two specific instances (TA002, TA008) is itself a caution against generalizing "ROSE is good at idle time" beyond this instance family without further testing.

Implementation-specific claims. The runtime, storage, and complexity results in Sections VI–VIII are properties of this specific Python/NumPy implementation, not architecture-independent proofs; a different implementation of the same mathematical model could shift the WT/WO runtime gap without changing the statistical ordering.

XII. Conclusion and Research Roadmap

ROSE (Relative Order SEquencer) learns the signed displacement between jobs, a representation that sits strictly between the edge view and the position view and reduces to each at a limit ($\Delta = +1$ for adjacency, $\text{sign}(\Delta)$ for order) without being a superset of either. Across a matched 20-seed ablation of six ROSE variants, template inheritance (WT) is a robust, statistically decisive winner over reconstruction from scratch (WO) — not just faster, but structurally different in its late-generation learning dynamics — and multi-reference averaging beats both single-reference alternatives, with the more expressive empirical histogram variant paying a real memory cost for expressiveness this sample size does not reward. Against the broader sixteen-algorithm field, ROSE/WT is mid-pack on makespan and total flow time and carries a real but statistically fragile, instance-concentrated signal on machine idle time.

The most defensible claim this paper makes is the family-internal one; the field-level claim is deliberately weaker, and "ROSE is the best permutation EDA" is explicitly not a claim this paper makes at any confidence level (Figure 12). What we take from the gap between these two claims is not a weakness to explain away but the paper's actual thesis: representation and objective interact, and no single fixed representation — ROSE included — should be expected to win uniformly. Longer budgets, tuned selection ratios, or an adaptive scheme that chooses among edge, position, and relational views per objective are the natural next steps; the last of these is the explicit design goal of CARE-EDA, this paper's planned successor.

Reproducibility

All six ROSE variants, both baseline representations, and every figure and table in this paper are generated by scripts in `scripts/`, from raw JSON archived by the original benchmark runs; exact reproduction commands, software versions, and known reproduction gaps are in `README.md` and `EXPERIMENT_GAPS.md`. No public code or data repository URL is committed as of this writing; this is a plain factual statement, not a placeholder, and is the author's decision to make before submission.

Acknowledgment

The author thanks the maintainers of the COINCIDENCE algorithm suite for the benchmark infrastructure this study builds on.

Fig. 12. Every headline claim in this paper, tagged with its evidentiary tier.

References

[1] S. Tsutsui, "Probabilistic model-building genetic algorithms in permutation representation domain using edge histogram," in *Parallel Problem Solving from Nature – PPSN VII*, ser. Lecture Notes in Computer Science, vol. 2439. Springer, 2002, pp. 224–233.

[2] ——, "Node histogram vs. edge histogram: A comparison of probabilistic model-building genetic algorithms in permutation domains," in *2006 IEEE Congress on Evolutionary Computation (CEC 2006)*, 2006, pp. 1939–1946.

[3] D. E. Goldberg and R. Lingle, "Alleles, loci, and the traveling salesman problem," in *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*. Pittsburgh, PA: Lawrence Erlbaum Associates, 1985, pp. 154–159.

[4] L. Davis, "Applying adaptive algorithms to epistatic domains," in *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85)*, vol. 1. Morgan Kaufmann, 1985, pp. 162–164.

[5] D. Whitley, T. Starkweather, and D. Fuquay, "Scheduling problems and traveling salesmen: The genetic edge recombination operator," in *Proceedings of the Third International Conference on Genetic Algorithms (ICGA 1989)*. Morgan Kaufmann, 1989, pp. 133–140.

[6] N. J. Radcliffe, "Forma analysis and random respectful recombination," in *Proceedings of the Fourth International Conference on Genetic Algorithms (ICGA 1991)*. San Diego, CA: Morgan Kaufmann, 1991, pp. 222–229.

[7] N. J. Radcliffe and P. D. Surry, "Fitness variance of formae and performance prediction," in *Foundations of Genetic Algorithms 3 (FOGA-95)*. Morgan Kaufmann, 1995, pp. 51–72.

[8] J. Ceberio, E. Irurozki, A. Mendiburu, and J. A. Lozano, "A review on estimation of distribution algorithms in permutation-based combinatorial optimization problems," *Progress in Artificial Intelligence*, vol. 1, no. 1, pp. 103–117, 2012.

[9] H. Mühlenbein and G. Paß, "From recombination of genes to the estimation of distributions I. Binary parameters," in *Parallel Problem Solving from Nature – PPSN IV*, ser. Lecture Notes in Computer Science, vol. 1141. Springer, 1996, pp. 178–187.

[10] P. Larrañaga and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, ser. Genetic Algorithms and Evolutionary Computation. Dordrecht: Kluwer Academic Publishers, 2002, vol. 2.

[11] M. Hauschild and M. Pelikan, "An introduction and survey of estimation of distribution algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 3, pp. 111–128, 2011.

[12] S. Tsutsui, M. Pelikan, and A. Ghosh, "Edge histogram based sampling with local search for solving permutation problems," *International Journal of Hybrid Intelligent Systems*, vol. 3, no. 1, pp. 11–22, 2006.

[13] S. Tsutsui, "Effect of using partial solutions in edge histogram sampling algorithms with different local searches," in *2009 IEEE International Conference on Systems, Man and Cybernetics (SMC 2009)*, 2009, pp. 2137–2142.

[14] M. A. Fligner and J. S. Verducci, "Distance based ranking models," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 48, no. 3, pp. 359–369, 1986.
[15] J. Ceberio, A. Mendiburu, and J. A. Lozano, "Introducing the mallows model on estimation of distribution algorithms," in *Neural Information Processing (ICONIP 2011)*, ser. Lecture Notes in Computer Science, vol. 7063. Springer, 2011, pp. 461–470.
[16] J. C. Bean, "Genetic algorithms and random keys for sequencing and optimization," *ORSA Journal on Computing*, vol. 6, no. 2, pp. 154–160, 1994.
[17] R. L. Plackett, "The analysis of permutations," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 24, no. 2, pp. 193–202, 1975.
[18] R. D. Luce, *Individual Choice Behavior: A Theoretical Analysis*. New York: John Wiley & Sons, 1959.
[19] J. Ceberio, A. Mendiburu, and J. A. Lozano, "The Plackett-Luce ranking model on permutation-based optimization problems," in *2013 IEEE Congress on Evolutionary Computation (CEC 2013)*, 2013, pp. 494–501.
[20] W. Wattanapornprom and P. Chongstitvatana, "Solving multimodal combinatorial puzzles with edge-based estimation of distribution algorithm," in *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation (GECCO Companion 2011)*, 2011, pp. 67–68.
[21] ——, "Negative correlation learning in the estimation of distribution algorithms for combinatorial optimization," *IEICE Transactions on Information and Systems*, vol. E96-D, no. 11, pp. 2397–2408, 2013.
[22] O. Srimongkolkul and P. Chongstitvatana, "Application of node based coincidence algorithm for flow shop scheduling problems," in *Proceedings of the 10th International Joint Conference on Computer Science and Software Engineering (JCSSE 2013)*, 2013, pp. 49–52, no DOI on record for this venue.
[23] W. Wattanapornprom, P. Olanviwitchai, P. Chutima, and P. Chongstitvatana, "Multi-objective combinatorial optimisation with coincidence algorithm," in *2009 IEEE Congress on Evolutionary Computation (CEC 2009)*, 2009, pp. 1675–1682.
[24] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
[25] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Tech. Rep. 103, 2001.
[26] M. Pelikan, K. Sastry, and D. E. Goldberg, "Multiobjective estimation of distribution algorithms," in *Scalable Optimization via Probabilistic Modeling*, ser. Studies in Computational Intelligence. Springer, 2006, vol. 33, pp. 223–248.
[27] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Modified distance calculation in generational distance and inverted generational distance," in *Evolutionary Multi-Criterion Optimization (EMO 2015)*, ser. Lecture Notes in Computer Science, vol. 9018. Springer, 2015, pp. 110–125.
[28] E. Taillard, "Benchmarks for basic scheduling problems," *European Journal of Operational Research*, vol. 64, no. 2, pp. 278–285, 1993.
[29] J. M. Framinan, R. Leisten, and R. Ruiz-Usano, "Efficient heuristics for flowshop sequencing with the objectives of makespan and flowtime minimisation," *European Journal of Operational Research*, vol. 141, no. 3, pp. 559–569, 2002, dOI not independently confirmed; metadata cross-checked via RePEc/IDEAS – see REFERENCE_GAPS.md.